

Contents

Preface	xiii
Prologue	1
1 Introduction	11
1.1 Introduction	12
1.2 Basic Theses	16
1.2.1 On Quantitative and Qualitative Perturbation	16
1.2.2 On Quantitative and Qualitative Equivalence	18
1.2.3 On The Stratification of Parameter Space	20
1.2.4 On Catastrophe Sets	22
1.2.5 On The Fragility of Degenerate Polynomials	23
1.2.6 On The Finite Word Length Problem	26
1.2.7 On The Existence of Fragile Polynomials and Matrices	31
1.2.8 On The Existence of Fragile Controllers	36
1.2.9 On the Fragility of ‘Resilient’ Optimal Controllers	40
1.2.10 On The Cost of Optimization	40
1.3 Fragility Issues in Control Systems	41
1.4 Objects in General Position and Genericity	47
1.5 Exceptional Objects and Degeneracies	51
1.6 Analysis of Feedback Systems	55
1.6.1 Closed Loop Models	60
1.6.2 Polynomial Methods	61
1.6.3 Matrix Methods	63
1.6.4 Plant Models	64
1.6.5 Controller Models	65
1.6.6 Coupled Plant-Controller Models	65
Exercises	68
References Cited	77

2	Catastrophes and Optimization	81
2.1	Elements of Catastrophe Theory	82
2.1.1	Catastrophes	82
2.1.2	Catastrophe Theory	83
2.1.3	Structural Stability	84
2.1.4	Forms and Morphogenesis	86
2.1.5	Degeneracies and Versal Deformation	87
2.1.6	Transversality	89
2.1.7	Bifurcation Set and Stratification of Parameter Space	89
2.1.8	Maxwell Set	91
2.1.9	Catastrophes in Differential Equations	92
2.1.10	Catastrophes in Integral Equations	94
2.1.11	Catastrophes in Polynomials	95
2.1.12	Catastrophes in Matrices	106
2.2	Optimization	110
2.2.1	Robust Unconstrained Optimization	111
2.2.2	Fragile Unconstrained Optimization	113
2.2.3	Constrained Optimization	116
	Exercises	121
	References Cited	129
3	Catastrophe Signatures	133
3.1	Versality and Transversality	134
3.1.1	Versality	135
3.1.2	Transversality	136
3.2	Versal Deformation of Polynomial Germs	137
3.2.1	Fragile Polynomials	140
3.2.2	Robust Polynomials	147
3.3	Loss of Transversality	152
3.3.1	Quadratic Polynomials	153
3.3.2	Cubic Polynomials	154
3.3.3	Quartic Polynomials	154
3.3.4	Quintic Polynomials	156
3.4	Degenerate Eigenvalues	158
3.5	Degenerate Decrement	159
3.6	Degenerate Resonance	163
3.7	Proximity to Catastrophe Set	164
3.8	Summary of Catastrophe Signatures	165
	Exercises	166
	References Cited	173

4	Catastrophes in Optimal Control	175
4.1	H_∞ Controller for Maximum Gain Margin	179
4.1.1	Reference Controller	179
4.1.2	Perturbed Controller	187
4.1.3	Alternative Full Order Implementation	195
4.1.4	Reduced Order Implementation	204
4.2	H_∞ Optimal Satellite Attitude Control	210
4.3	H_∞ Optimal Control of DC Motor	217
4.4	Optimal Control of X-29 Aircraft	222
4.5	H_2 Control for Minimum Sensitivity	225
4.6	LQR Optimal Controller	229
4.7	“Resilient” Controller	231
4.8	H_∞ Control with Full State Feedback	240
	Exercises	247
	References Cited	258
5	Structurally Stable Controllers	263
5.1	Introduction	264
5.2	PID Controller	265
5.3	Lead-Lag Network Controllers	267
5.4	Bridge-T Network Compensator	270
5.5	Pole Placement Design	271
5.6	Exception	273
	Exercises	275
	References Cited	277
6	Real and Complex Singularities	279
6.1	Introduction	280
6.1.1	Real Algebraic Curves	281
6.1.2	Complex Cases	281
6.1.3	Novel Representations	281
6.1.4	Versality and Transversality	281
6.2	Test for Structural Stability	282
6.3	Real and Complex Algebraic Curves	283
6.3.1	Critical Points	287
6.3.2	Versality and Transversality	287
6.4	Square Root Mapping	288
6.4.1	Analog Form of the Square Root Mapping	288
6.4.2	Complimentary Square Root Mapping	294
6.4.3	Digital Form of the Square Root Mapping	294

6.5	Diagrammatic Representation	295
6.6	Conformal Mapping	296
6.7	Contour Mapping	301
6.8	Vector Field Representation	303
6.9	Domain Coloring	310
	Exercises	314
	References Cited	316
7	Matrix Methods	319
7.1	Introduction	320
7.2	Matrix Realizations of Transfer Functions	321
7.2.1	Controller Canonical Form	321
7.2.2	Observer Canonical Form	325
7.2.3	Other Realizations	326
7.3	Similarity Transformation	328
7.3.1	Non-transversal Intersections of Orbits	329
7.3.2	Effect of Small Perturbations in General	333
7.3.3	Small Perturbations via Finite Word Length	336
7.4	Fragile Matrices and Fragile Controllers	340
7.4.1	Optimal Controller Designs	340
7.4.2	Optimal Filter Designs	342
7.5	Robust Matrices	349
7.6	Concluding Remarks	352
	Exercises	353
	References Cited	365
8	Catastrophe-Theoretic Methods	367
8.1	Normal Forms	368
8.2	Cuspoid Normal Forms	370
8.2.1	Derivative of Potential Function	371
8.2.2	Proportional of Potential Function	371
8.2.3	Integral of Potential Function	371
8.2.4	General Considerations	372
8.3	Stratification of Parameter Space by Catastrophe Sets	373
8.3.1	The Fold Catastrophe	374
8.3.2	The Cusp Catastrophe	376
8.3.3	The Swallow-Tail Catastrophe	382
8.3.4	The Butterfly Catastrophe	389
8.4	Proximity to Catastrophe Sets	397
	Exercises	398

<i>CONTENTS</i>	xi
References Cited	406
Epilogue	409
Bibliography	411
Index	421

Preface

This book of mine has little need of preface, for indeed it is 'all preface' from beginning to end.

D'ARCY W. THOMPSON
On Growth and Form

Catastrophes occur in a control system when a small perturbation of the parameters of the controller leads to a considerable change in the qualitative properties of the closed loop system. Such a controller is said to be “fragile.” Optimal controllers are particularly fragile. A fragile controller is, indeed, a “structurally unstable” controller.

The fragility or structural instability of controllers is a serious matter that deserves the attention of researchers, practitioners and students of automatic control systems. This book is written to provide guidance in this direction. The primary focus is on optimal controllers, because they are structurally unstable by design. We apply the twin concepts of *versal deformation* and *transversality* which are suitable for investigating structural instability phenomena. The subject matter of structural stability originates from the works of Henri Poincaré, and was significantly expounded by René Thom. Structural stability considerations are indispensable when investigating the dynamic behavior of various mathematical objects, if the objects depend on parameters which are subject to small perturbations in the course of time.

The problem of controller fragility was first brought to public attention a few years ago when two researchers observed that some optimal control systems were easily destabilized by small perturbations. The controllers studied

by the investigators exhibited such an intolerant degree of sensitivity to small perturbations that they were tagged ‘fragile controllers.’ Although the results published by the investigators are readily verifiable, it is not universally accepted by control theorists at present that such fragility is related to optimization. Indeed, many procedures to ‘fix’ the fragility problem have been proposed in the optimal control literature. None of the proposed remedies, however, addresses the fundamental cause of fragility directly. In view of the fact that a considerable body of work on optimal control is extant, it seems desirable to provide an explanation for the kind of fragility that has been reported, and to provide a context for its understanding. Such a context is provided by catastrophe theory—a theory for investigating the *geometric change in form* of various mathematical objects under smooth parameter variation.

Catastrophe theory, as a branch of mathematics, originated from the works of the mathematician René Thom in the late 1960s and early 1970s. Key postulates of Thom were formulated as theorems and proved by Mather, Malgrange, and others. Significant extensions and generalizations were made by V. I. Arnol’d. E. C. Zeeman also made landmark contributions to the mathematics and applications of the theory. It is a matter of historical record that some controversies were generated during the early, formative years of catastrophe theory. The controversies were not about the mathematical validity of catastrophe theory itself, but on certain of its applications in the social sciences. Even during the sharpest controversies, the theoretical principles embodied in the mathematics of catastrophe theory were not in dispute, nor its applications in the ‘exact’ sciences. In particular, when catastrophe theory was applied in physics and engineering, it was generally found to yield new facts or illuminate previously known results from another perspective. Thus, the mathematics of catastrophe theory itself, and its applications in various fields of engineering, are irreproachably founded.

There is an intimate relationship between catastrophe theory on one hand, and linear control theory on the other. In order to see the relationship clearly, one may note that the *characteristic polynomial* widely encountered in linear control theory is diffeomorphic to the generalized cusp catastrophe, otherwise known as *cuspid catastrophe*, or the A_k *singularity*. Moreover, the entire edifice of ‘state-space’ linear systems theory is built with matrices. The stability of matrices with parametrically varying elements has been examined by Arnol’d* using catastrophe-theoretic methods. Thus, polynomials and matrices both lie at the intersection of catastrophe theory and control theory.

From what is known about optimization procedures on one hand, and the

*V. I. Arnol’d: “On Matrices Depending on Parameters.” *Russian Mathematical Surveys*, vol. 26, pp. 29–43, 1971.

transversality theorems of catastrophe theory on the other, it should not be a surprise that optimal controllers are inherently ‘fragile.’ One does not have to search intensively before coming across clear and unmistakable symptoms of structural instability in optimal controllers. If control theorists and engineers seem to be unaware of catastrophes intrinsically lurking in optimal control designs, then the situation may be compared to an army going to battle oblivious of the existence of land mines in the battlefield. It may turn out at war-end that all the troops have survived unwounded. Nevertheless, a prudent general would prefer to have mine-detection capabilities in advance. This book is written in an effort to provide some methods and means to enable those who so desire to have the capability of detecting potential catastrophes in optimal controller designs.

In order to understand catastrophes and fragility problems in optimal controllers, it is useful to start with an examination of fragile matrices and polynomials. In chapter 1, certain prevalent conceptions in control theory concerning fragile matrices and polynomials are discussed, and the basic theses of this book laid out. After that, the polynomials and matrices of a typical feedback system are treated as geometric objects, paying particular regard to their qualitative properties. The idea of objects being in ‘general position’ or being ‘exceptional’ is then discussed. The concepts of ‘structural stability’ and ‘catastrophes’ are examined in chapter 2. This opens up our introduction to catastrophe theory and the closely related singularity theory, as qualitative methods of mathematical analysis suitable for investigating optimized systems. It is shown that whereas some optimization methods may be unimodal, leading to models that are robust, others may create models that are multi-modal, and hence highly unstable or fragile under arbitrary perturbation. The significance of loss of transversality in constrained optimization is highlighted. In chapter 3, certain tell-tale signs attendant upon catastrophes in control systems are presented. By such signs or signatures, one may ascertain at the design stage whether or not a contemplated controller would have a tendency to be fragile, or whether it would be truly robust.

Selected designs from the optimal control literature are examined in chapter 4. It is shown that the polynomials appearing in the transfer functions of a typical, optimized controller show symptoms of fragility, such as non-transversality at intersections and various kinds of degeneracies. The optimal controller designs examined include those that are based on H_∞ , H_2 and \mathcal{L}^1 methods. The polynomials of other types of controllers which have not been designed using optimization methods, such as classical PID controllers, are discussed in chapter 5 for comparison. Various ways of depicting the geometry of controller polynomials are surveyed in Chapter 6, wherein a brief

Prologue

The same local situation can give birth to apparently different outcomes under the influence of unknown or unobservable factors.

R. THOM

Structural Stability and Morphogenesis

Catastrophes occur in computational processes when degenerate polynomials and matrices are encountered. In this regard, a catastrophe is manifest when a small perturbation changes the *qualitative properties* of the computed solutions completely. In many cases, we are not even aware of the perturbation imposed by the computing apparatus, but it is there nevertheless. A perturbation is always present whether we are computing with embedded controllers using only a few bits, or with sophisticated 32- or 64-bit machines running a well-tested computational program such as MATLAB which is widely used in classical and optimal control studies. As will be shown presently, the problem actually solved by the computer is not always the same problem, *qualitatively*, that the user thinks he or she has specified. In the same way, the controller implemented in a given instance is not always qualitatively the same as the theoretical controller the designer thinks he or she has specified.

Effect of Unavoidable Perturbation

The polynomials and matrices used in theoretical designs of various controllers are not the same polynomials and matrices that emerge from software and hardware implementations. The actual controller is a perturbation of the theoretical controller, because *a theoretical design can never be implemented exactly*. Consequently, the controller used in a practical setting is always under the influence of “unknown or unobservable” perturbations. The effect of the perturbations may be ignored in many cases, but not when we are dealing with degenerate or nearly degenerate polynomials and matrices. We illustrate the problem by showing that the unknown perturbations imposed by the computational apparatus can alter, in a qualitatively significant manner, the solution properties of polynomials and matrices in the neighborhood of degeneracy.

Consider, for example, the following characteristic polynomial of an undamped vibrating system,

$$f(\lambda) = \lambda^6 - 12\lambda^5 + 59\lambda^4 - 152\lambda^3 + 216\lambda^2 - 160\lambda + 48 = 0.$$

The roots of the polynomial, $\lambda = \omega^2$, are the eigenvalues of the undamped system, where ω is the oscillation frequency. All the eigenvalues must be real and positive if the solution is to be admissible. One may open up a MATLAB window and type the following line of code

```
roots([1, -12, 59, -152, 216, -160, 48])
```

The output returned by the computer may look like that shown below:

```
ans =
    3.0000
    2.0004 + 0.0004i
    2.0004 - 0.0004i
    1.9996 + 0.0004i
    1.9996 - 0.0004i
    1.0000
```

Qualitatively, the result is in error. The correct solutions are all real and positive, since the polynomial may be factored as

$$f(\lambda) = (\lambda - 3)(\lambda - 2)(\lambda - 2)(\lambda - 2)(\lambda - 2)(\lambda - 1) = 0.$$

Even when using this factored form, and a strong hint is given to the computer that the roots of f are all real, MATLAB fails to return qualitatively accurate solutions, as the following transcript shows.

```
>> format long, roots(poly([3,2,2,2,2,1]))

ans =

    3.000000000000051
    2.00040451759297 + 0.00040488761398i
    2.00040451759297 - 0.00040488761398i
    1.99959548240675 + 0.00040414782175i
    1.99959548240675 - 0.00040414782175i
    1.000000000000005
```

The persistent appearance of complex roots continues to render the solutions for eigenvalues and natural frequencies *qualitatively* inadmissible.

Evidently, something is wrong somewhere. Why is MATLAB, one of the most rigorously tested and frequently used computational packages in engineering unable to correctly compute, qualitatively, the roots of a low-order polynomial to more than 4 decimal places, even when using double precision? How much worse would a lowly, embedded computer in an optimal controller fare when exposed to the harsh environmental realities of *in situ* applications?

In considering the above questions, and others such like, it is important to understand that the computer does not actually solve the problem posed, but another one infinitesimally close to it. Whenever we feed a polynomial or matrix into a computer or controller design, the polynomial or matrix automatically comes under the influence of unknown perturbations imposed by the software and hardware. The qualitative consequence of the perturbation is sometimes of considerable significance. Instead of the specified polynomial, f , the computer sees a nearby polynomial,

$$\begin{aligned} \bar{f}(\lambda) = & \lambda^6 - (12 + \epsilon_1)\lambda^5 + (59 + \epsilon_2)\lambda^4 - (152 + \epsilon_3)\lambda^3 + (216 + \epsilon_4)\lambda^2 \\ & - (160 + \epsilon_5)\lambda + (48 + \epsilon_6) = 0. \end{aligned}$$

where $\epsilon_1 \dots \epsilon_6$ are perturbation parameters. The same occurs when controllers in an automatic control system are implemented in practice. Generally, one has almost no clue as to the values of the perturbation parameters imposed upon the theoretical design by the software and hardware deployed in the implementation. No matter how elegant the algorithm, or the degree of sophistication to which the hardware is built, the controller actually implemented imposes an unavoidable amount of perturbation upon the nominal design. It turns out that while the effects of the unknown perturbations may be benign

and negligible in certain cases, they can also be “catastrophic” in certain other instances.

The effect of the unknown perturbation introduced by the computational apparatus is benign in the case of the following characteristic polynomial for another undamped vibrating system,

$$g(x) = x^6 - 11.8x^5 + 56.75x^4 - 142.25x^3 + 195.75x^2 - 139.95x + 40.5 = 0.$$

In this case, MATLAB is able to compute the solutions accurately to more than 10 decimal places, as shown in the following record.

```
>> format long
>> roots([1,-11.8,56.75,-142.25,195.75,-139.95,40.5])

ans =

    2.99999999999971
    2.50000000000071
    1.99999999999957
    1.79999999999973
    1.50000000000030
    0.99999999999998
```

The exact solutions may be deduced from the factored form of the polynomial,

$$g(x) = (x-1)(x-1.5)(x-1.8)(x-2)(x-2.5)(x-3).$$

This fact is confirmed by displaying the result in short form, i.e. to four decimal places, as the transcript below shows.

```
>> format short
>> roots([1,-11.8,56.75,-142.25,195.75,-139.95,40.5])

ans =

    3.0000
    2.5000
    2.0000
    1.8000
    1.5000
    1.0000
```

The foregoing is a demonstration of the fact that some polynomials such as $f(\lambda)$ are fragile, while others such as $g(x)$ are robust. Under the influence of unknown perturbation imposed by the computational hardware and software, the fragile polynomials yield results that are qualitatively different from the true solutions. In contrast, robust polynomials are relatively insensitive to the same kinds of perturbation.

To round up this introductory sampling of fragile and robust polynomials, let us carry out a few numerical experiments centered on the Wilkinson polynomial. It is convenient to use the following template to generate a polynomial, the roots of which are the first few natural numbers. Let the basic formula be written as

$$q_n = \prod_{i=1}^n (s-i) = (s-1)(s-2)\cdots(s-n-1)(s-n),$$

so that

$$q_4 = \prod_{i=1}^4 (s-i) = (s-1)(s-2)(s-3)(s-4),$$

the Wilkinson polynomial is

$$q_{20} = \prod_{i=1}^{20} (s-i) = (s-1)(s-2)\cdots(s-20),$$

and

$$q_{30} = \prod_{i=1}^{30} (s-i) = (s-1)(s-2)\cdots(s-30).$$

The polynomial q_4 is robust under the algorithm used by MATLAB, as the transcript below shows.

```
>> q4=poly(1:4); roots(q4)

ans =

    4.0000
    3.0000
    2.0000
    1.0000
```

However, if we try to compute the roots of q_{30} using MATLAB, we obtain non-trivial complex components. This observation is supported by the following transcript.

```
>> q30=poly(1:30); roots(q30)
```

```
ans =
```

```
32.6106  
31.7272 + 2.9361i  
31.7272 - 2.9361i  
29.3612 + 5.3387i  
29.3612 - 5.3387i  
26.1263 + 6.7832i  
26.1263 - 6.7832i  
22.6657 + 7.1859i  
22.6657 - 7.1859i  
19.4321 + 6.7626i  
19.4321 - 6.7626i  
16.6111 + 5.8218i  
16.6111 - 5.8218i  
14.1888 + 4.6097i  
14.1888 - 4.6097i  
13.8364  
12.0708 + 3.2861i  
12.0708 - 3.2861i  
10.2544 + 1.9785i  
10.2544 - 1.9785i  
8.8408 + 0.6768i  
8.8408 - 0.6768i  
8.0125  
6.9819  
6.0022  
4.9999  
4.0000  
3.0000  
2.0000  
1.0000
```

Thus, the root of largest modulus has been moved from 30 to 32.6106, which is somewhat of interest, quantitatively. However, of greater interest is the qualitative point of view: the *real* roots of magnitudes 28 and 29 have, presumably, been transformed into a *complex* conjugate pair, $31.7272 \pm j 2.9361$.

This qualitatively significant transformation of real quantities into entities with imaginary components has been made by the computing machinery without any warning or notice whatsoever to the user. As would be made manifest in later chapters of this book, this notable event is aided and abetted by various kinds of *degeneracies*, some obvious, others somewhat latent.

At any rate, the foregoing shows that although we may feed the polynomial $q_i(s)$ into the computer, the computer actually extracts the roots of a different polynomial, $\bar{q}(s)$. This latter polynomial is a perturbation of the polynomial we think we are working with, but the perturbation is unknown to us. In certain circumstances, the effect of the unknown perturbation may have huge qualitative significance upon the solutions, even when we are using the finest computing apparatus and working in double or treble precision.

In contrast to the foregoing cases in which the perturbation was not under our control and, in a sense, unknown to us, let us consider the following examples in which we can specify the perturbation to some extent. It should be appreciated, however, that a residual amount of unknown perturbation is still superimposed upon the known perturbation that we may explicitly specify.

Suppose the nominal state matrix of a certain controller design is A_0 , but the hardware and software combine to give us the matrix $A_1 = A_0 + \delta A$ or, perhaps, the matrix $A_2 = A_0 - \delta A$, in the course of implementation. Let us affix values to the nominal matrix and the perturbation matrix as follows,

$$A_0 = \begin{bmatrix} -128 & -27 & 3 \\ 653.0 & 138 & -15 \\ 287 & 63 & -4 \end{bmatrix}, \quad \delta A = \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

The extremely small size of the perturbation matrix, δA , relative to that of the nominal matrix, A_0 , should not escape the attention of the reader.

In accordance with the previously stated formulas, the perturbed matrices are

$$A_1 = \begin{bmatrix} -128 & -27 & 3 \\ 653.1 & 138 & -15 \\ 287 & 63 & -4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -128 & -27 & 3 \\ 652.9 & 138 & -15 \\ 287 & 63 & -4 \end{bmatrix}.$$

It may be verified by the reader that the eigenvalues of A_0 returned by MATLAB are, to 10 decimal places, the numbers 1, 2, 3. However, as the following transcript shows, the eigenvalues of A_1 are significantly different from the eigenvalues of the nominal matrix, in a qualitative sense.

```
>> A1=[-128,-27,3;653.1,138,-15;287,63,-4]

A1 =

-128.0000 -27.0000  3.0000
 653.1000 138.0000 -15.0000
 287.0000  63.0000 -4.0000

>> eig(A1)

ans =

1.500000000000041 + 1.56524758425156i
1.500000000000041 - 1.56524758425156i
2.999999999999925
```

A comparison of the eigenvalues of the nominal matrix A_0 with those of its perturbations, A_1 and A_2 , is displayed in the following table. On the basis of the data in the table, one may claim that the nominal matrix A_0 is fragile with respect to eigenvalue computation.

eigenvalues of A_0	eigenvalues of A_1	eigenvalues of A_2
0.99999999999542	1.500000000000041 + 1.56524758425156i	-0.21755640372862
2.00000000000741	1.500000000000041 - 1.56524758425156i	3.21755640372781
2.99999999999714	2.99999999999925	3.00000000000068

On the other hand, the nominal matrix B_0 below is relatively robust with respect to eigenvalue computation, when similarly small changes are made in the matrix elements. The matrix and its perturbations are defined as follows,

$$B_1 = B_0 + \delta B, \quad B_2 = B_0 - \delta B,$$

where

$$B_0 = \begin{bmatrix} -128 & -27 & 3 \\ 653.0 & 138 & -15 \\ 287 & 63 & -24 \end{bmatrix}, \quad \delta B = \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$B_1 = \begin{bmatrix} -128 & -27 & 3 \\ 653.1 & 138 & -15 \\ 287 & 63 & -24 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -128 & -27 & 3 \\ 652.9 & 138 & -15 \\ 287 & 63 & -24 \end{bmatrix}.$$

The eigenvalues are computed by using MATLAB. A comparison of the results is shown in the table below. One may claim that, in this case, the qualitative character of the eigenvalues is preserved under the perturbation.

eigenvalues of B_0	eigenvalues of B_1	eigenvalues of B_2
10.38238017426804	10.20629583373915	10.55424536181147
-2.99999999999953	-2.84412133847187	-3.15154263910442
-21.38238017426852	-21.36217449526734	-21.40270272270710

It is clear from the foregoing that some polynomials such as $f(\lambda)$ are fragile when exposed to perturbation, while others such as $g(x)$ are robust. In the same way, some matrices such as A_0 are fragile, while others such as B_0 are robust. The fragility of $f(\lambda)$ and A_0 is due to *intrinsic degeneracy*. The degeneracy is obvious in f but not in A_0 . We propose to show in the remainder of this book that certain controllers—such as optimal controllers—are *inherently fragile* due to precisely the same reason: intrinsic degeneracy. A controller with intrinsic degeneracy is not robust, in a qualitative sense, under perturbation. In the course of implementing a controller, a certain amount of perturbation is unavoidable. The perturbation is always present, ‘unknown or unobservable.’ Although the perturbation may be extremely small, we cannot ignore it altogether because, *qualitatively*, it can sometimes give rise to catastrophes in the control system.

1

Introduction

One of the central problems studied by mankind is the problem of succession of form. . . . The purpose of science is to foresee this change of form and, if possible, explain it.

R. THOM

Structural Stability and Morphogenesis

Matrices and polynomials are the main building blocks of control theory. In order to understand fragile controllers, it is advantageous to begin with a study of fragile polynomials and matrices. Usually, the algebraic characteristics of these mathematical objects are studied in control theory, and mainly from the quantitative viewpoint. However, matrices and polynomials also have qualitative properties and *geometric forms*, and may be studied as geometric objects. Most geometric objects are ‘in general position,’ and thus have ‘generic’ properties. Others have ‘exceptional’ properties, and are generally characterized by *degeneracy*. Objects in general position have stable forms, while those that are exceptional have extremely fragile forms. Accordingly, fragile polynomials and matrices—which are not in general position—undergo drastic changes in form under arbitrary perturbation. Since fragile matrices and polynomials are designed by default into optimal controllers, one may, by means of catastrophe-theoretic reasoning, foresee the impending fragile behavior, and attempt an explanation for it, as suggested by Thom.

Suppose p_0 is the characteristic polynomial of a closed loop system, and p_1 is a perturbation of p_0 , such that $p_1 = p_0 - \epsilon$, and $|\epsilon| \approx 0$ or $\|\epsilon\| \approx 0$, where

$$p_0(s) = s^8 + 85s^7 + 1311s^6 + 14404s^5 + 78031s^4 + 192330s^3 + 234369s^2 + 139320s + 32400, \quad (1.3)$$

and

$$\epsilon(s) = 0.0988s^8 + 0.0583s^7 + 0.0423s^6 + 0.0516s^5 + 0.0334s^4 + 0.0433s^3 + 0.0226s^2 + 0.0580s + 0.0760. \quad (1.4)$$

Accordingly, $p_1 = p_0 - \epsilon$ is given by

$$p_1(s) = 0.9012s^8 + 84.9417s^7 + 1310.9577s^6 + 14403.9484s^5 + 78030.9666s^4 + 192329.9567s^3 + 234368.9774s^2 + 139319.9420s + 32399.9240. \quad (1.5)$$

The coefficients of the above perturbation polynomial, $\epsilon(s)$, were generated using a pseudo random number generator. It may be verified by comparing the 2-norm, or any suitable norm preferred by the reader, that

$$\frac{\|\epsilon\|}{\|p_0\|} \approx 0.$$

For example, using the following MATLAB script,

```
format long;
p0=[1,58,1311,14404,78031,192330,234369,139320,32400];
e=[.0988,.0583,.0423,.0516,.0334,.0433,.0226,.0580,.0760];
p1=p0-e;
%
n0=norm(p0,2)
n1=norm(p1,2)
ne=norm(e,2)
```

we obtain

$$\|p_0\|_2 = 344496.0286041045, \quad \|p_1\|_2 = 344495.9485567557,$$

so that

$$\|\epsilon\|_2 = 0.17393501660103 \quad \Rightarrow \quad \frac{\|\epsilon\|}{\|p_0\|} = 5.04897 \times 10^{-7} \approx 0.$$

Evidently, we can make a valid claim that the perturbation in this case is quantitatively insignificant, since $\|\epsilon\|/\|p_0\| \approx 0$. However, does this mean that the effect of the ‘‘insignificant’’ perturbation, ϵ , may be written off?

Certainly not, and in support of this answer, let us draw and compare the graphs of p_0 and p_1 . When we do so, we get the illustrations displayed as Figures 1.2(a) and (b) respectively.

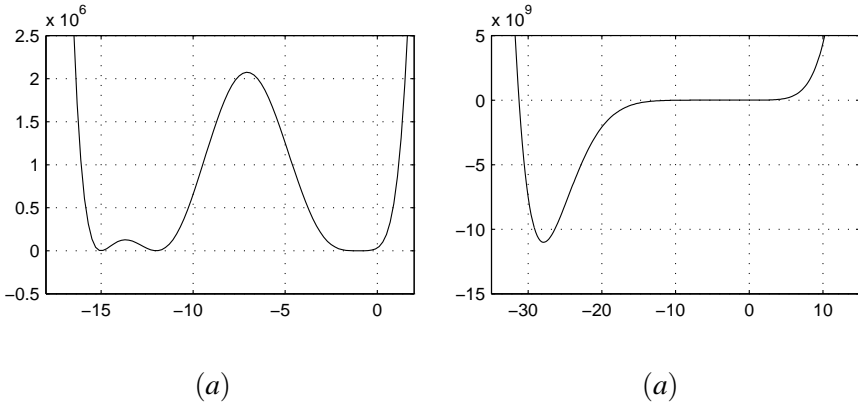


FIGURE 1.2 Graph of (a) the nominal polynomial p_0 defined in Equation (1.3), and (b) the polynomial $p_1 = p_0 - \epsilon$ defined in Equation (1.5). The dramatic change in shape resulting from the perturbation, which is evidenced by the change in the number and nature of the critical points and critical values of the nominal polynomial, may be noted in the above graphs.

From a qualitative point of view, the effect of this ‘small’ perturbation is certainly not negligible; the perturbation has induced a considerable distortion of the geometric form of the nominal polynomial, p_0 . Although the perturbation is quantitatively insignificant, it cannot be summarily written off by a careful investigator who is interested in qualitative behavior. Thus, *a negligible quantitative perturbation is not necessarily a negligible qualitative perturbation.*

1.2.2 On Quantitative and Qualitative Equivalence

II. *Two polynomials may be extremely close quantitatively and yet be very far apart qualitatively. The same is also true of matrices and controllers.*

The idea that a generalized mathematical object G_1 is equivalent to another object G_2 if the two of them are ‘‘close’’ in some metric is pervasive in control theory. It is, however, not always right. Indeed, it is the Achilles’ heel of optimal control. A normed linear space (e.g. a Hilbert space) is the main

polynomial equations. The roots of a polynomial $p(s)$ are determined from the intersections of the polynomial with the s -axis. The slope p' of the polynomial may intersect with the s -axis, in the real* case, in one of the ways depicted in Figure 1.11. In the illustrations, the dashed line represents a segment of the s -axis, while the solid line represents a close-up view of a patch of the polynomial in the neighborhood of the s -axis. The intersections depicted in (a)–(c) are in general position. The exceptional case corresponds to an intersection at zero slope, $p' = 0$, and is shown in (d).

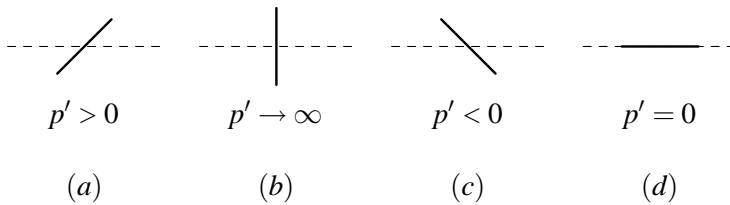


FIGURE 1.11 Local structure of real intersections showing various types. Type (d) is not in general position.

What has been said above admits of a matrix representation. Thus, one may write

$$\mathbf{Ax} = \mathbf{b},$$

and speak of \mathbf{A} being ‘well-’ or ‘ill-conditioned’ with respect to solving for the intersection point \mathbf{x} , given the right hand side \mathbf{b} . A matrix that is well-conditioned with respect to solving simultaneous equations or, equivalently, with respect to matrix inversion, may yet be ill-conditioned with respect to solving the eigenproblem

$$\mathbf{Ax} = s\mathbf{x}.$$

Here s is the eigenvalue parameter, and the set of eigenvalues is obtained from the roots of the polynomial $|s\mathbf{I} - \mathbf{A}| = 0$.

If we consider the problem of matrix inversion, there is a property that ‘almost all’ invertible matrices must have. Such a property—that the matrix be nonsingular—is called a *generic property*. Non-singularity in this context is determined by the determinant of the matrix. Thus, if the determinant of any square matrix \mathbf{A} does not vanish, then one can be ‘almost certain’ that the matrix is invertible. The exceptional or non-generic case arises only when the determinant vanishes, i.e. $|\mathbf{A}| = 0$. The generic property of $|\mathbf{A}| \neq 0$ applies to solving $\mathbf{Ax} = \mathbf{b}$, but not to the eigenproblem $\mathbf{Ax} = s\mathbf{x}$. In this latter case,

*Complex intersections are discussed later, in Chapter 6.

3.1 Versality and Transversality

Two of the most important concepts in using catastrophe theory for engineering applications are *versality* and *transversality*. In the particular case with which we are concerned, i.e. the investigation of robustness and fragility issues in optimal control systems, the twin concepts of versality and transversality are very instructive. The general procedure when using these concepts in applications is as follows. First, one settles the problem of versality, after which one examines the transversality properties at intersections.

Versality concerns *deformation* or *distortion* of the geometric shape of an object as a result of perturbation. If the object is a polynomial, we first obtain its shape in the unperturbed state. After that, we observe any deformation of the shape of the graph as the perturbation is applied.

In the case of transversality, we first select a fixed subspace of the ambient space in which the primary object of interest is smoothly embedded. For example, if we are interested in the transversality properties that govern the roots at the intersection of a polynomial $p(s)$ with the s -axis, then the ambient space is the plane on which a graph of the polynomial is drawn, while the fixed submanifold is the s -axis. Both this axis and the polynomial itself are smoothly embedded in the plane. If even *one* of the intersections of the polynomial with the s -axis is not transverse, then it means there is residual structural instability somewhere in the system.

Fragile controllers are closely related to fragile matrices and fragile polynomials. Because of this close relationship, we can recognize fragile controllers by studying the characteristics of fragile matrices and fragile polynomials. In order for the control engineer to have the capability of predicting the fragility or otherwise of a contemplated controller design, it is useful to have the capability of identifying fragile polynomials and matrices.

It turns out that only a few special kinds of polynomials and matrices are fragile. Indeed, almost all polynomials and matrices are in general position. They are thus ‘robust’ or ‘resilient’ under perturbation. The exceptional matrices and polynomials are those that are not in general position. Consequently, one effective means of identifying fragile controllers is to look for polynomials and matrices that are not in general position. The proven way to do this is by a skillful application of *transversality*, as has been done by Thom [3.1] and Arnol’d [3.2]. This is the essence of their qualitative method upon which catastrophe theory is built. Transversality is a generic property. Accordingly, it must hold “nearly always”. In the rare cases when it fails to hold, we must pay attention because a break down of transversality signifies that a considerable qualitative change has occurred, or is about to occur.

At the 1997 American Automatic Control Conference in Albuquerque, New Mexico, Keel and Bhattacharyya [4.1] presented their results of a detailed investigation of the problem of fragility in optimal control systems. When their results were first announced, a number of control theory specialists expressed surprise, even disbelief. However, the few in the audience who were familiar with catastrophe theory instantly recognized one of the fingerprints of the cusp catastrophe that was clearly present in the data presented. In particular, one notices immediately the ‘near degeneracies’ in the poles of the closed loop system presented as Example 1 in [4.1], and reproduced in the left column of the following table:

<i>Closed loop poles to 4 decimal places</i>	<i>Closed loop poles to 3 decimal places</i>
$-0.4666 \pm j \ 14.2299$	$-0.467 \pm j \ 14.230$
$-5.5334 \pm j \ 11.3290$	$-5.533 \pm j \ 11.330$
-1.0002	-1.000
$-1.0000 \pm j \ 0.0002$	$-1.000 \pm j \ 0.000$
-0.9998	-1.000

As may be observed from the above data, the poles of the closed loop system do have a *fourfold degeneracy* at $s = -1$ when the data are presented to 3 decimal places. This is a classical fingerprint of the cusp catastrophe. It indicates that a cusp catastrophe germ is hidden somewhere in the closed loop system. We may also repeat the calculations by using symbolic computation or integer arithmetic, e.g. using MAPLE (or, Maxima*) instead of floating point arithmetic in MATLAB (or, Octave). The closed loop poles are then found to be, to 4 decimal places,

$$\begin{aligned}
 & -0.4666 \pm j \ 14.2299 \\
 & -5.5334 \pm j \ 11.3290 \\
 & \quad -1.0000 \\
 & \quad -1.0000 \\
 & \quad -1.0000 \\
 & \quad -1.0000
 \end{aligned}$$

*Maxima is a freely obtainable symbolic computation package, distributed under the terms of the GNU Public License, while Octave is a free version of matrix computational tools, also distributed under the same GNU Public license. For more information, you may point your browser at the following urls which are known to be active as at the time of writing: <http://www.octave.org> or <http://www.ma.utexas.edu/maxima.html>.

of the corresponding polynomial for the nominal system shown in Figure 4.5 on page 186. The actual, numerical values of the roots of Equation (4.8) are obtained by computation with MAPLE, and are as displayed in Table 4.11. These values may be compared to the zeros of the reference controller tabulated on page 187. As in the reference case, there is a cluster of roots near $s = -1$.

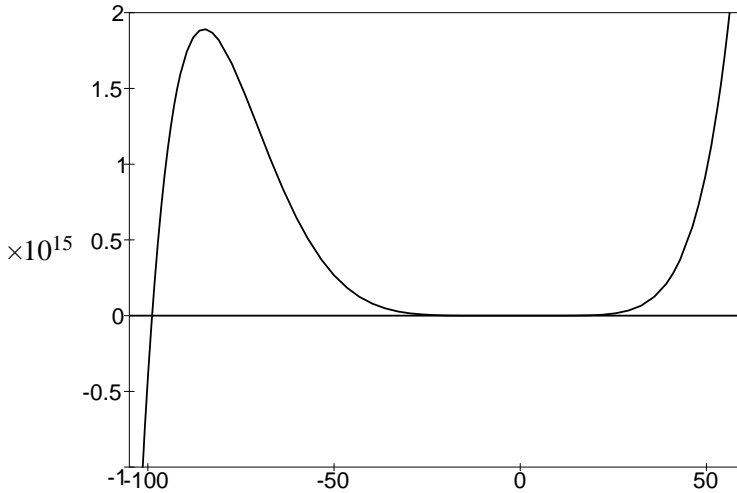


FIGURE 4.9 Numerator of the closed loop transfer function of the perturbed system. This is a highly unstable polynomial, with non-transversal intersections near $s = -1$. The structural instability of the closed loop is inherited from the optimal controller inserted in the loop.

We now turn to the denominator of the closed loop transfer function of the perturbed system. The closed loop poles are listed in the second column of Table 4.11. It may be observed from Table 4.11 that two of the poles have now migrated to the right hand side of the s -plane, leading to a loss of (temporal) stability of the control system. Moreover, the degeneracy of roots is not completely unfolded yet, and there remains other pairs of pathologically close eigenvalues in the neighborhood of $s = -1$. These eigenvalues may yet collide, leading to “modal coalescence” and subsequent bifurcation of modes. Such collisions generally occur between a pair of eigenvalues. In some cases, some of the modes may escape to the right hand side of the s -plane after the eigenvalue collision, especially if the collision is between eigenvalues of “opposite signatures”.

The graph of the closed loop characteristic polynomial of the perturbed system is shown in Figure 4.9. This graph may be compared to that of the

4.1.4 Reduced Order Implementation

Several “order reduction” techniques are often used in optimal controller design procedures to reduce the degree of controller polynomials. For example, in Reference [4.3], there are three roots in the neighborhood of $s = -1$ that are common to both numerator and denominator of the controller. The three roots are canceled out of the sixth order controller to arrive at a third order controller. The reduced order controller has the transfer function which may be written in factored form as

$$C(s) = 149.97 \frac{(s+1)(s+1.028)(s+100)}{(s+1.855)(s+70.74)(s-205.97)}, \quad (4.11)$$

while the transfer function of the plant is, in factored form,

$$P(s) = \frac{s-1}{(s+1)(s-2)}. \quad (4.12)$$

The numerator of the reduced order controller is now a third order polynomial, and is shown in Figure 4.15. It may be seen that even this low-order polynomial, which is a cubic, has a non-transversal intersection with the s -axis near $s = -1$. The optimization algorithm has made loss of transversality an intrinsic part of this optimal controller.

It may also be noted that there is a “pole-zero cancellation” problem in this control system. The factor $(s+1)$ occurs in both the numerator of the controller’s transfer function in Equation (4.11), and the denominator of the plant’s transfer function in Equation (4.12). Thus, the plant’s pole at $s = -1$ is “cancelled” by the controller’s zero at the same location.

The denominator of the reduced order controller’s transfer function of Equation (4.11) is graphed in Figure 4.16. This polynomial is now structurally stable, as evident from all three real intersections in Figure 4.16. Each intersection of the cubic with the s -axis is transverse, and in general position.

The transfer function of the closed loop system may be written in two forms: first, as T_1 , without pole-zero cancellation, and secondly as T_2 with the cancellation. Let

$$T_1(s) = \frac{N_1(s)}{D_1(s)} \quad (4.13)$$

where

$$\begin{aligned} N_1(s) &= (s-1)(s+1)(s+1.028)(s+100) \\ D_1(s) &= (s+1)(s-2)(s+1.855)(s+70.74)(s-205.97) \\ &\quad + 149.97(s+1)(s+1.028)(s+100)(s-1), \\ &= s^4 + 14.595s^3 + 446.77971s^2 + 2880.146221s + 38638.96304. \end{aligned}$$